**1. Understanding Flume**

Flume is a top-level project at the Apache Software Foundation. While it can function as a general-purpose event queue manager, in the context of Hadoop it is most often used as a log aggregator, collecting log data from many diverse sources and moving them to a centralized data store.

**Flume Components**

A Flume data flow is made up of five main components: Events, Sources, Channels, Sinks, and Agents:

**Events** An event is the basic unit of data that is moved using Flume. It is similar to a message in JMS and is generally small. It is made up of headers and a byte-array body.

**Sources** The source receives the event from some external entity and stores it in a channel. The source must understand the type of event that is sent to it: an Avro event requires an Avro source.

**Channels** A channel is an internal passive store with certain specific characteristics. An in-memory channel, for example, can move events very quickly, but does not provide persistence. A file-based channel provides persistence. A source stores an event in the channel where it stays until it is consumed by a sink. This temporary storage lets source and sink run asynchronously.

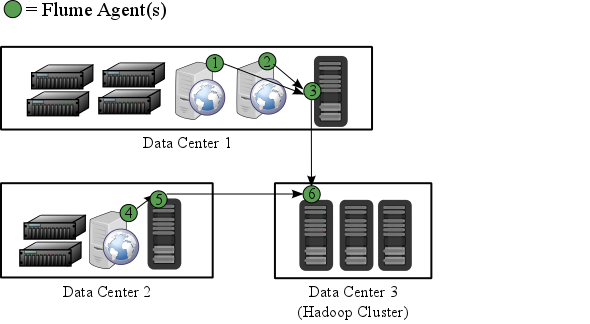
**Sinks** The sink removes the event from the channel and forwards it to either to a destination, like HDFS, or to another agent/dataflow. The sink must output an event that is appropriate to the destination.

**Agents** An agent is the container for a Flume data flow. It is any physical JVM running Flume. An agent must contain at least one source, channel, and sink, but the same agent can run multiple sources, sinks, and channels. A particular data flow path is set up through the configuration process.

**2. Installing Flume**

Flume is included in the HDP repository, but it is not installed automatically as part of the standard HDP installation process. Hortonworks recommends that administrators not install Flume agents on any node in a Hadoop cluster. The following image depicts a sample topology with six Flume agents:

* Agents 1, 2, and 4 installed on web servers in Data Centers 1 and 2.
* Agents 3 and 5 installed on separate hosts in Data Centers 1 and 2 to collect and forward server data in Avro format.
* Agent 6 installed on a separate host on the same network as the Hadoop cluster in Data Center 3 to write all Avro-formatted data to HDFS



**Prerequisites**

1. You must have at least core Hadoop on your system. See [Configure the Remote Repositories](http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.6/bk_installing_manually_book/content/config-remote-repositories.html) for more information.
2. Verify the HDP repositories are available:

yum list flume

The output should list at least one Flume package similar to the following:

flume.noarch 1.5.2.2.2.6.0-2800.el6 HDP-2.3

If yum responds with "Error: No matching package to list" as shown below, yum cannot locate a matching RPM. This can happen if the repository hosting the HDP RPMs is unavailable, or has been disabled. Follow the instructions at [Configure the Remote Repositories](http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.6/bk_installing_manually_book/content/config-remote-repositories.html) to configure either a public or private repository before proceeding.

Error: No matching package to list.

1. You must have set up your JAVA\_HOME environment variable per your operating system. See [JDK Requirements](http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.3.6/bk_installing_manually_book/content/meet-min-system-requirements.html#jdk-requirements-getting-started) for instructions on installing JDK.

export JAVA\_HOME=/path/to/java

1. The following Flume components have HDP component dependencies. You cannot use these Flume components if the dependencies are not installed.

**Table 17.1. Flume 1.5.2 Dependencies**

| **Flume Component** | **HDP Component Dependencies** |
| --- | --- |
| HDFS Sink | Hadoop 2.3 |
| HBase Sink | HBase 0.98.0 |
| Hive Sink | Hive 0.13.0, HCatalog 0.13.0, and Hadoop 2.3 |

**Installation**

Verify the HDP repositories are available for your Flume installation by entering yum list flume. See Prerequisites for more information.

To install Flume from a terminal window, type:

* For RHEL or CentOS:

yum install flume

yum install flume-agent #This installs init scripts

* For SLES:

zypper install flume

zypper install flume-agent #This installs init scripts

* For Ubuntu and Debian:

HDP support for Debian 6 is deprecated with HDP 2.3.6. Future versions of HDP will no longer be supported on Debian 6.

apt-get install flume

apt-get install flume-agent #This installs init scripts

The main Flume files are located in /usr/hdp/current/flume-server. The main configuration files are located in /etc/flume/conf.

**3. Configuring Flume**

To configure a Flume agent, edit the following three configuration files:

* flume.conf
* flume-env.sh
* log4j.properties

**flume.conf**

Configure each Flume agent by defining properties in a configuration file at /etc/flume/conf/flume.conf. The init scripts installed by the flume-agent package read the contents of this file when starting a Flume agent on any host. At a minimum, the Flume configuration file must specify the required [sources](https://flume.apache.org/FlumeUserGuide.html#flume-sources), [channels](https://flume.apache.org/FlumeUserGuide.html#flume-channels), and [sinks](https://flume.apache.org/FlumeUserGuide.html#flume-sinks) for your Flume topology.

For example, the following sample Flume configuration file defines a [NetCat Source](https://flume.apache.org/FlumeUserGuide.html#netcat-source), [a Memory Channel](https://flume.apache.org/FlumeUserGuide.html#memory-channel) and a [Logger Sink](https://flume.apache.org/FlumeUserGuide.html#logger-sink). This configuration lets a user generate events and subsequently logs them to the console.

# example.conf: A single-node Flume configuration

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = netcat

a1.sources.r1.bind = localhost

a1.sources.r1.port = 44444

# Describe the sink

a1.sinks.k1.type = logger

# Use a channel that buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel

a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1

This configuration defines a single agent named a1. a1 has a source that listens for data on port 44444, a channel that buffers event data in memory, and a sink that logs event data to the console. The configuration file names the various components, and describes their types and configuration parameters. A given configuration file might define several named agents.

See the [Apache Flume User Guide](https://flume.apache.org/FlumeUserGuide.html) for a complete list of all available Flume components.

To see what configuration properties you can adjust, a template for this file is installed in the configuration directory at /etc/flume/conf/flume.conf.properties.template.

A second template file exists for setting environment variables automatically at startup:

/etc/flume/conf/flume- env.sh.template.

|  |  |
| --- | --- |
|  | **Note** |
| If you use an [HDFS Sink](https://flume.apache.org/releases/content/1.4.0/FlumeUserGuide.html#hdfs-sink), be sure to specify a target folder in HDFS. |

**flume-env.sh**

Set environment options for a Flume agent in /etc/flume/conf/flume-env.sh:

* To enable JMX monitoring, add the following properties to the JAVA\_OPTS property:
* JAVA\_OPTS="-Dcom.sun.management.jmxremote
* -Dcom.sun.management.jmxremote.port=4159
* -Dcom.sun.management.jmxremote.authenticate=false

-Dcom.sun.management.jmxremote.ssl=false"

* To customize the heap size, add the following properties to the JAVA\_OPTS property:

JAVA\_OPTS= "-Xms100m -Xmx4000m"

**log4j.properties**

Set the log directory for log4j in /etc/flume/conf/log4j.properties:

flume.log.dir=/var/log/flume

**4. Starting Flume**

There are two options for starting Flume.

* To start Flume directly, run the following command on the Flume host:

/usr/hdp/current/flume-server/bin/flume-ng agent -c /etc/flume/conf -f /etc/flume/conf/ flume.conf -n agent

* To start Flume as a service, run the following command on the Flume host:

service flume-agent start

## 5. HDP and Flume

Flume ships with many source, channel, and sink types. The following types have been thoroughly tested for use with HDP:

**Sources**

* Exec (basic, restart)
* Syslogtcp
* Syslogudp

**Channels**

* Memory
* File

**Sinks**

* HDFS: secure, nonsecure
* HBase

## 6. A Simple Example

The following snippet shows some of the kinds of properties that can be set using the properties file. For more detailed information, see the “Flume User Guide.”

agent.sources = pstream

agent.channels = memoryChannel

agent.channels.memoryChannel.type = memory

agent.sources.pstream.channels = memoryChannel

agent.sources.pstream.type = exec

agent.sources.pstream.command = tail -f /etc/passwd

agent.sinks = hdfsSinkagent.sinks.hdfsSink.type = hdfs

agent.sinks.hdfsSink.channel = memoryChannel

agent.sinks.hdfsSink.hdfs.path = hdfs://hdp/user/root/flumetest

agent.sinks.hdfsSink.hdfs.fileType = SequenceFile

agent.sinks.hdfsSink.hdfs.writeFormat = Text

The source here is defined as an exec source. The agent runs a given command on startup, which streams data to stdout, where the source gets it.

In this case, the command is a Python test script. The channel is defined as an in-memory channel and the sink is an HDFS sink.